Grad Project – Recipe Site

Team Member

Swapnil Vijay Gaidhankar (803012319)

CPSC 473-02 Web Programming and Data Management Spring, 2016

Prof: Kenytt Avery Department of Computer Science California State University, Fullerton

May 2, 2016

Introduction:

Recipe Site
An application to post and view recipes posted by users.

Functionality:
1. User is able to add recipe.
2. User is able to view recipe.

Application requirements and Installation steps:

Requirements:

- AngularJs
- Hapi Js
- Mongo Db
- Node version 4 or grater. and node-modules- , express, path, twitter.
- Module installed: Inert, Path, MongoDb

Installation steps:

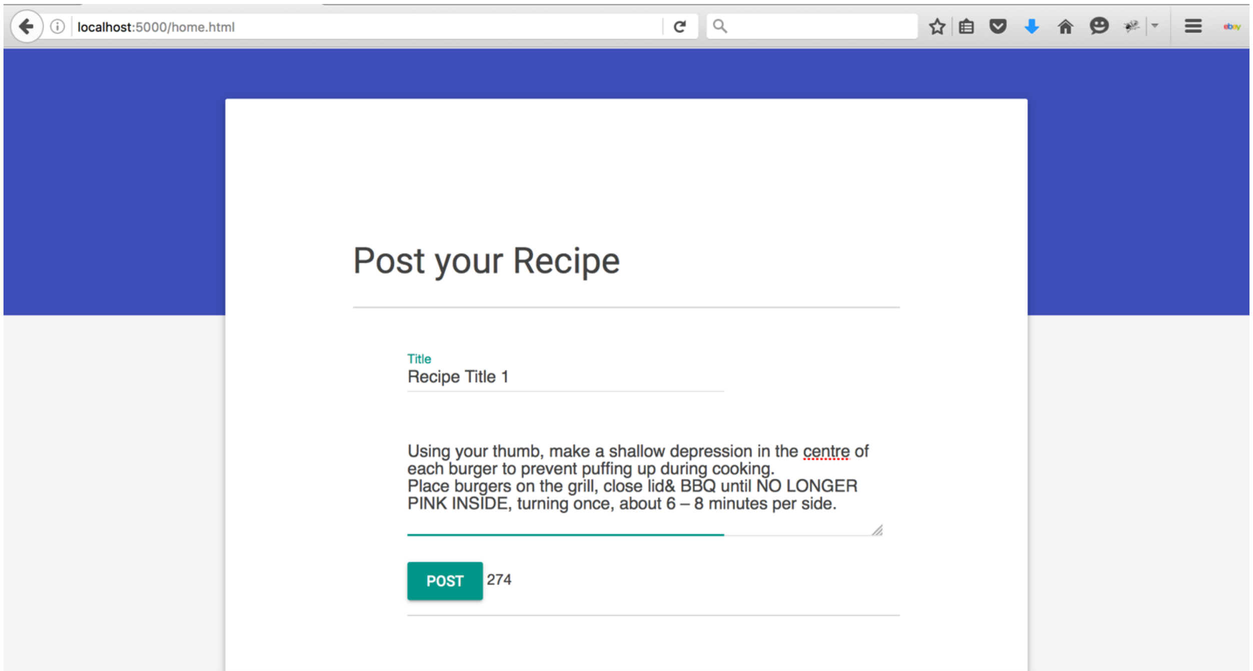
- Download and Install Node - https://nodejs.org/en/download/
- Install required node modules.
  - Hapi - `npm install –save hapi`
  - Path - `npm install –save path`
  - MongoDb - `npm install hapi-mongodb`
  - inert- `npm install inert`
- Start mongoDb on URL: mongodb://localhost:27017/test (Using test as database)
- Start the node server in appropriate directory where **server.js** is placed.
  Type command ‘node .’ to start server.
- Hit the URL: http://localhost:5000/home.html

Functional and Technical Deails:

Post Recipe

Functional Details:
Any user can able to post recipe following constrain:

1. Post title must be grater than 10 characters and less than 30 characters.
2. Post text must be grater than 10 characters and less than 500 characters.



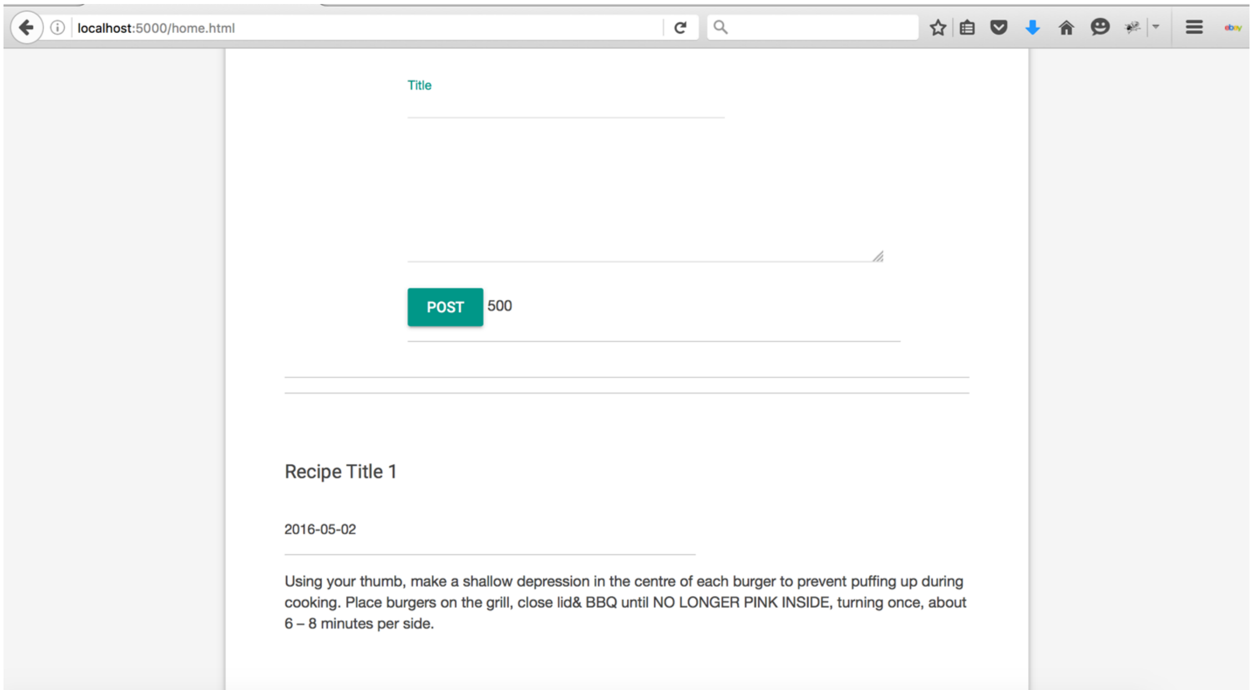
Post your Recipe

Title

Recipe Title 1

Using your thumb, make a shallow depression in the centre of each burger to prevent puffing up during cooking.
Place burgers on the grill, close lid& BBQ until NO LONGER PINK INSIDE, turning once, about 6 – 8 minutes per side.

274



Title

500

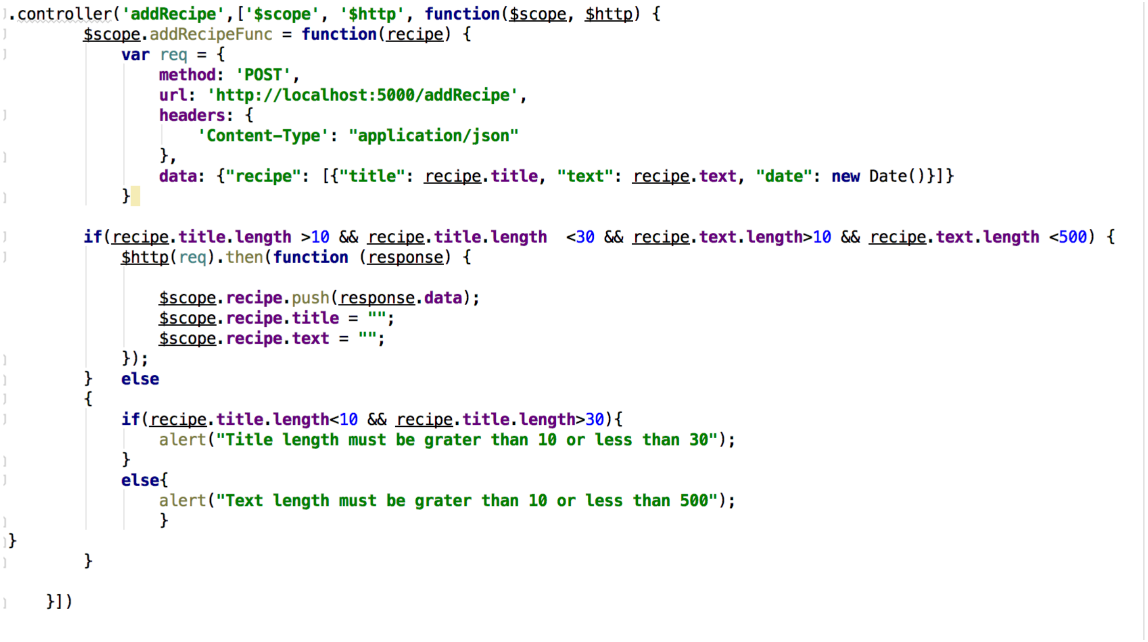
Recipe Title 1

2016-05-02

Using your thumb, make a shallow depression in the centre of each burger to prevent puffing up during cooking. Place burgers on the grill, close lid& BBQ until NO LONGER PINK INSIDE, turning once, about 6 – 8 minutes per side.

Code:

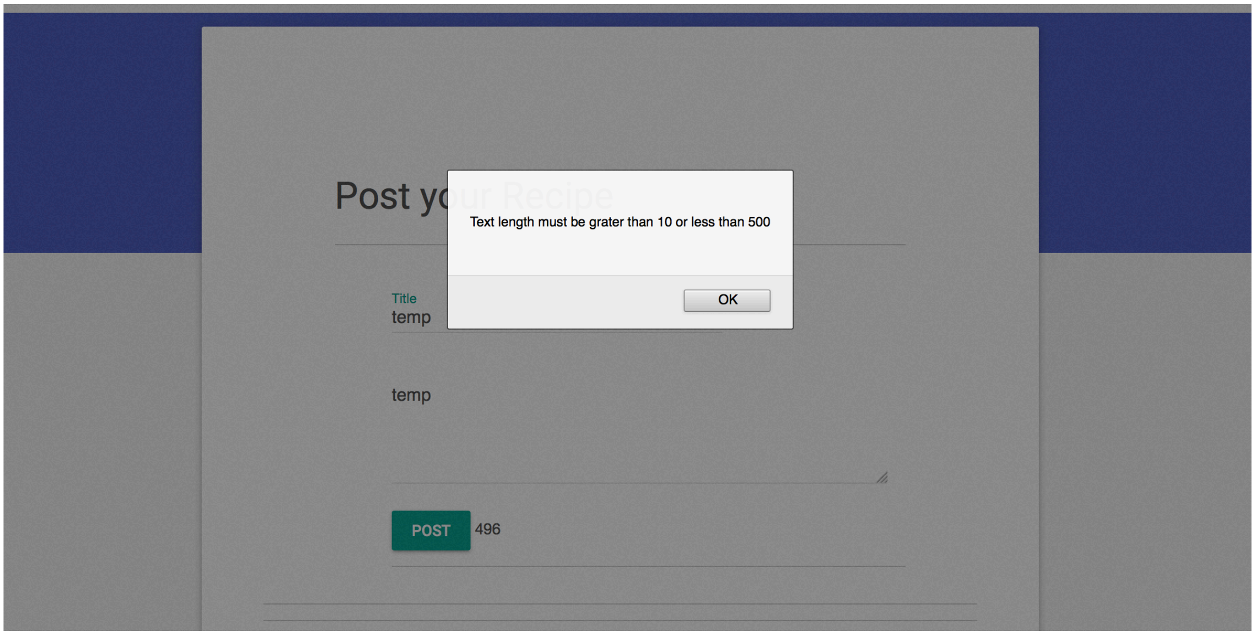
- AngularJS:



```
.controller('addRecipe',['$scope', '$http', function($scope, $http) {
        $scope.addRecipeFunc = function(recipe) {
            var req = {
                method: 'POST',
                url: 'http://localhost:5000/addRecipe',
                headers: {
                    'Content-Type': "application/json"
                },
                data: {"recipe": [{"title": recipe.title, "text": recipe.text, "date": new Date()}]}
            }

        if(recipe.title.length >10 && recipe.title.length  <30 && recipe.text.length>10 && recipe.text.length <500) {
            $http(req).then(function (response) {

                $scope.recipe.push(response.data);
                $scope.recipe.title = "";
                $scope.recipe.text = "";
            });
        }   else
        {
            if(recipe.title.length<10 && recipe.title.length>30){
                alert("Title length must be grater than 10 or less than 30");
            }
            else{
                alert("Text length must be grater than 10 or less than 500");
                }
}
        }

    }])
```

Constrains:

1. Post title must be grater than 10 characters and less than 30 characters.
2. Post text must be grater than 10 characters and less than 500 characters.



Post your Recipe

Text length must be grater than 10 or less than 500

Title
temp

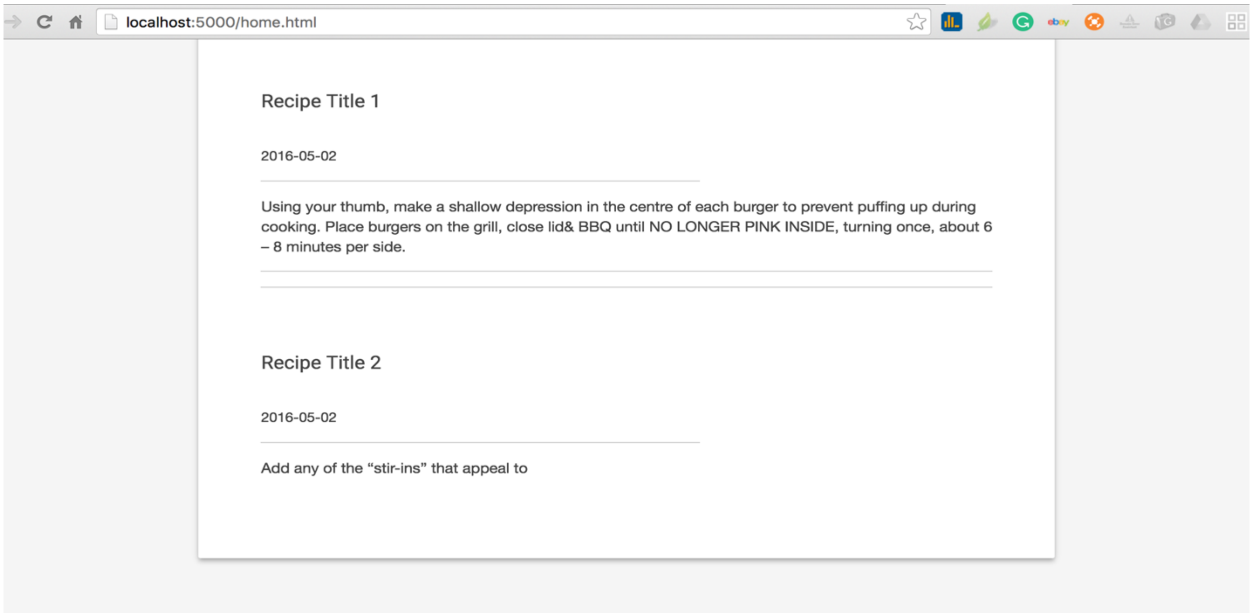
temp

496

Get Recipe

Functional Details:

User is able to see all the recipes which are posted till date once visit the website.



Recipe Title 1

2016-05-02

Using your thumb, make a shallow depression in the centre of each burger to prevent puffing up during cooking. Place burgers on the grill, close lid& BBQ until NO LONGER PINK INSIDE, turning once, about 6 – 8 minutes per side.

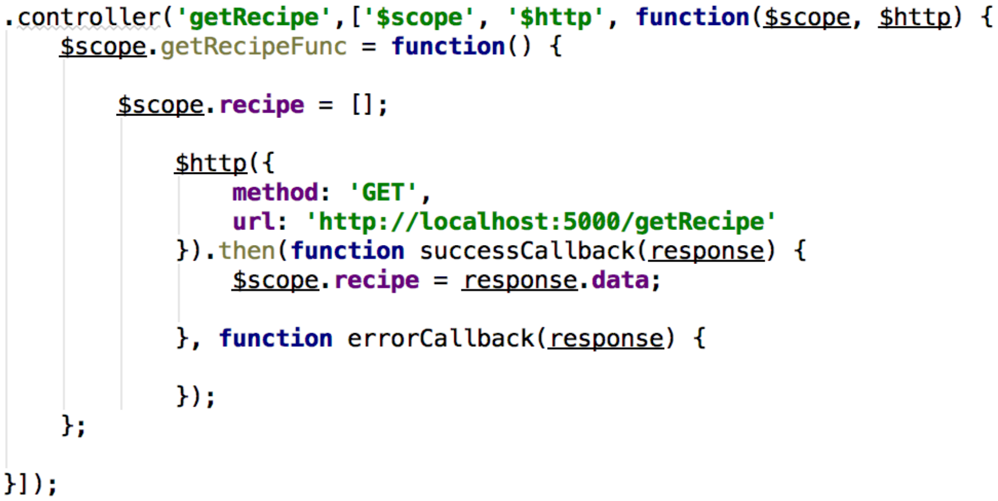
Recipe Title 2

2016-05-02

Add any of the “stir-ins” that appeal to

Code:

- AngularJS:



```
.controller('getRecipe',['$scope', '$http', function($scope, $http) {
    $scope.getRecipeFunc = function() {

        $scope.recipe = [];

            $http({
                method: 'GET',
                url: 'http://localhost:5000/getRecipe'
            }).then(function successCallback(response) {
                $scope.recipe = response.data;

            }, function errorCallback(response) {

            });
    };

}]);
```

File Structure:

- **server.js:** HapiJS server receive request from angular client
- **home.html:** Html file for recipes representations.
- **home.js:** client model which sends data received from home.html.

- **home.css:** style sheet file to decorate home.html.

References:

- https://www.npmjs.com/package/hapi-mongodb
- https://www.npmjs.com/package/path
- https://www.npmjs.com/package/inert
- http://hapijs.com/tutorials/getting-started
- https://angularjs.org/